Interactive OpenCV Introduction

Manipulate images with clojure and opencv

```
(ns spacial-dusk
  (:require
    [opencv3.core :refer :all]
    [opencv3.utils :as u]))
```

```
nil
```

```
(def neko
  (u/mat-from-url "https://s-media-cache-
ak0.pinimg.com/236x/10/2c/75/102c756d7e808deff666f3edf540abba.jpg"))
(u/mat-view neko)
```

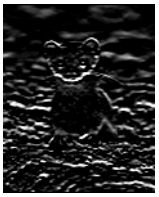
```
(def gray (new-mat))
(cvt-color neko gray COLOR_RGB2GRAY)
(u/mat-view gray)
```

```
(def small-gray-neko
  (u/resize-by gray 0.5))
(u/mat-view small-gray-neko)
```

```
(def equalized (new-mat))
(equalize-hist small-gray-neko equalized)
(u/mat-view equalized)
```

```
(def dilated (u/mat-from small-gray-neko))
(def dilation-size 2)
(def element (get-structuring-element  MORPH_RECT
  (new-size (inc (* 2 dilation-size)) (inc (* 2 dilation-size)))))
(dilate small-gray-neko dilated element)
(u/mat-view dilated)
```

```
(def sobelled (u/mat-from small-gray-neko))
(sobel small-gray-neko sobelled -1 0 1)
(u/mat-view sobelled)
```



```
(def bit_not (u/mat-from small-gray-neko))
(bitwise-not small-gray-neko bit_not)
(u/mat-view bit_not)
```